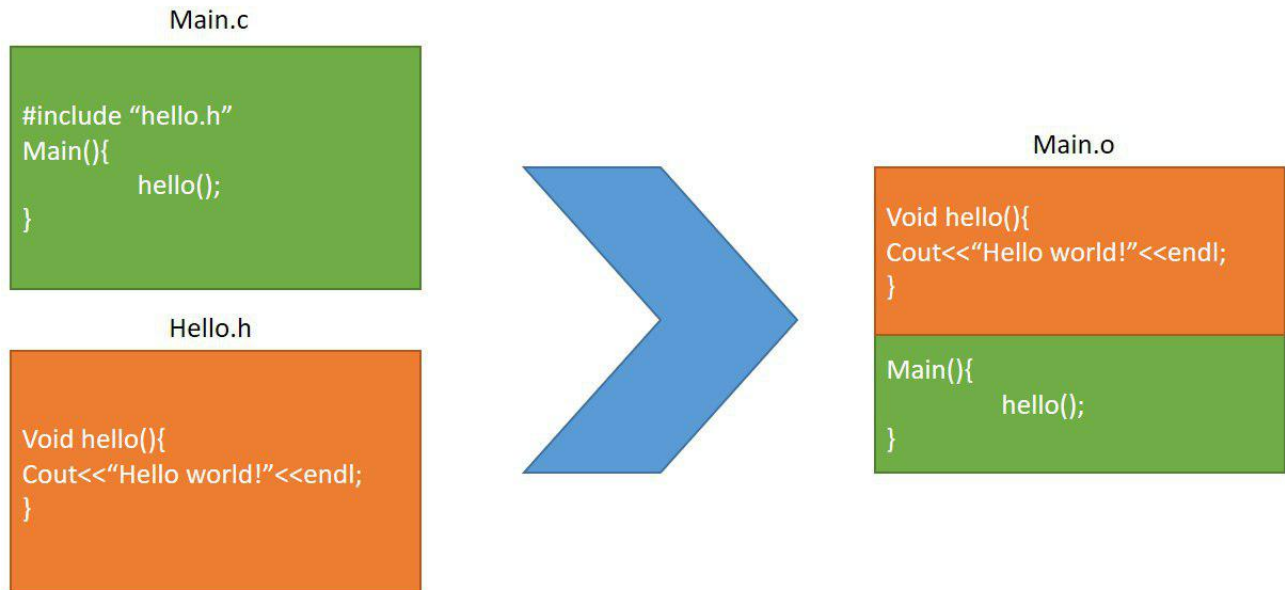


Tarea 2: Expansión

Profesores: Nelson Baloian
Patricio Poblete

Auxiliares: Gabriel Azócar
Manuel Cáceres
Michel Llorens
Sergio Peñafiel

Fecha de Entrega: 21 de Abril 23:59hrs



1 Introducción

En la computación, muchas veces al escribir un programa (independiente del lenguaje) uno hace referencia a otros códigos o funciones que suelen estar directamente en otros archivos. Sin embargo, copiar y pegar todo este código al escribir un programa no es conveniente y lo vuelve poco legible.

Por tanto, generalmente los lenguajes compilados como lo son Java, C++ y C permiten utilizar referencias a otros archivos, las cuales al momento de compilar se **expanden** completando con el

código correcto.

En C, la palabra clave de la referencia es `#include <file>`, la cual va sola en una línea y le permite al compilador rellenar ese espacio con el archivo `file`.

2 Explicación

2.1 Expandir

Para esta tarea, la idea es crear un programa que funcione tal y como un compilador, es decir, vaya leyendo un archivo y por facilidad, imprimiéndolo en pantalla (utilizando la salida estándar) con todos los reemplazos correspondientes. En nuestro caso las referencias serán denotadas con la forma `<<<nombre_archivo>>>`. Un ejemplo de esto es.

Archivo 1: archivo1

```
El archivo 2 dice:\n
<<<archivo2>>>, y el archivo 3\n
dice: <<<archivo3>>>\n
aquí termina el archivo 1.\n
```

Archivo 2: archivo2

```
[contenido del archivo 2]\n
```

Archivo 3: archivo3

```
El archivo 3 contiene al\n
archivo 4 aquí: <<<archivo4>>>\n
```

Archivo 4: archivo4

```
[contenido del archivo 4]\n
```

Nota: el char `\n` es un salto de línea (o newline) y no se ve en un archivo y tampoco son dos caracteres distintos.

Ahora, el resultado de pasar el archivo1 por nuestro programa de expansión debería mostrar en la salida lo siguiente:

```
El archivo 2 dice:\n
[contenido del archivo 2]\n
, y el archivo 3\n
dice: El archivo 3 contiene al\n
```

```
archivo 4 aqui:[contenido del archivo 4]\n\n\naqui termina el archivo 1.\n
```

Por términos de eficiencia el programa debería ser capaz de trabajar con un listado de archivos y así no tener que operar de a uno. Por tanto si se le entregan como argumento archivo2 y archivo4 (en ese orden), debería imprimir:

```
[contenido del archivo 2]\n[contenido del archivo 4]\n
```

3 Implementación

En esta tarea usted deberá entregar un archivo `Expand.java` que recibirá como argumentos de ejecución un listado de nombres de archivos, a los cuales realizará de forma **recursiva** el proceso de expansión.

Se debe tener particular cuidado con las referencias circulares entre los mismos archivos, como lo son $archivo1 \Rightarrow archivo2 \Rightarrow archivo3 \Rightarrow archivo1$, ya que estas producirán indudablemente un ciclo infinito de llamadas recursivas y el programa se caerá. Por tanto usted debe detectar estas situaciones y detenerse (evitando así el `StackOverflowException`). En caso contrario su programa debería terminar normalmente.

La forma en que se ejecutará su programa será desde línea de comando y para ejemplificar, a continuación una ejecución con 3 archivos (podrían ser N):

```
> java Expand archivo1 archivo2 archivo3
```

Por facilidad, los archivos se deben encontrar en la misma carpeta que el código.

4 Reglas

- Esta tarea debe ser resuelta en Java.
- Es obligatorio la entrega de un informe en formato pdf junto con su tarea (Ver siguiente sección).
- Esta tarea es de carácter individual, cualquier caso de copia se evaluará con la nota mínima.
- No olvide subir a U-cursos **todos** los archivos necesarios para que su tarea funcione correctamente.
- Debe subir los archivos de código fuente (*.java). Los archivos compilados (*.class) no serán evaluados.
- Cualquier duda respecto a la tarea puede ser consultada usando el foro del curso.
- No se aceptan atrasos.

5 Informe

El informe debe describir el trabajo realizado, el código fuente desarrollado, los resultados obtenidos y las conclusiones o interpretaciones de estos. Principalmente sea conciso, describiendo cada uno de los puntos que a continuación se indican.

- **Portada:** Indicando número de la tarea, fecha, autor, email, código del curso.
- **Introducción:** Descripción breve del problema y su solución.
- **Análisis del problema:** Exponga en detalle el problema, los supuestos que pretende ocupar, casos de borde y brevemente la metodología usada para resolverlo.
- **Solución del problema:**
 - Algoritmos de solución, incluyendo toda la información y figuras que considere necesarias.
 - Partes relevantes del código fuente
 - Ejemplos de entradas y salidas escogidos por usted.
- **Modo de uso:** explicando brevemente cualquier dato extra necesario para la compilación y ejecución de su programa.
- **Resultados y análisis:** Todo el análisis de los resultados, los gráficos, imágenes y la discusión requerida.